

# *NextBestOnce*: Achieving Polylog Routing despite Non-greedy Embeddings

Stefanie Roos, Thorsten Strufe  
 TU Darmstadt, CASED  
 firstname.lastname@cased.de

Social Overlays suffer from high message delivery delays due to insufficient routing strategies. Limiting connections to device pairs that are owned by individuals with a mutual trust relationship in real life, they form topologies restricted to a subgraph of the social network of their users. While centralized, highly successful social networking services entail a complete privacy loss of their users, Social Overlays at higher performance represent an ideal private and censorship-resistant communication substrate for the same purpose.

Routing in such restricted topologies is facilitated by embedding the social graph into a metric space. Decentralized routing algorithms have up to date mainly been analyzed under the assumption of a perfect lattice structure. However, currently deployed embedding algorithms for privacy-preserving Social Overlays cannot achieve a sufficiently accurate embedding and hence conventional routing algorithms fail. Developing Social Overlays with acceptable performance hence requires better models and enhanced algorithms, which guarantee convergence in the presence of local optima with regard to the distance to the target.

We suggest to model Social Overlays as graphs embedded in  $\mathbb{Z}_n^m$  with a scale-free degree distribution with exponent  $\alpha$ . The inaccuracy of the embedding is measured by a parameter  $C$ . We then show that our previously introduced routing algorithm *NextBestOnce* achieves an expected routing length of  $\mathcal{O}(\log^{\alpha-1} n \log \log n + C^3 \log n)$  on our Social Overlay model. A lower bound on the performance of *NextBestOnce* is given by  $\Omega(\log^{\alpha-1} n + C)$ . Furthermore, we show that leveraging information from the two-hop neighborhood, a Neighbor-of-Neighbor (NoN) modification of our algorithm achieves an expected routing length of  $\mathcal{O}(\log^{\delta(\alpha)(\alpha-1)} n \log \log n + C^3 \log n)$ , where  $\delta(\alpha) < 1$ . Hence, NoN information can indeed be used to improve the asymptotic routing complexity by more than a constant factor.

## 1 Introduction

Centralized communication platforms, such as online social networking (OSN) services, concentrate data and control in one point. Delivering all messages and published content through the centralized provider, they allow for perfect tracking and tracing of the communicating individuals. Current, highly successful services, like for instance Facebook, prohibit encryption and hence gain full access to the exchanged content. To enhance reliability, user-control, and privacy of communication, several decentralized approaches have been suggested recently.

The extent of decentralization of such platforms varies, depending on their trust assumptions and security objectives: hybrid schemes of decentralized servers, like diaspora<sup>1</sup>, replace the single centralized instance by several interconnected servers to which the users register and connect. Though the chosen servers can still monitor messages and behavior of their users, no central entity has full access to all data. Further decentralization is sought by peer-to-peer OSNs [1, 6], which fully decentralize the service provision to all participating parties. The devices of all participants, henceforth called *nodes*, are interconnected in an overlay that allows for

---

<sup>1</sup><http://www.joindiaspora.com>

content discovery, publication, and retrieval. Access control in these systems is enforced through encryption and key management. Participating in the peer-to-peer overlay, the nodes accept and establish connections with arbitrary other nodes, thus disclosing their network address to strangers, which potentially crawl the network to discover and monitor the participation of individuals. *Social Overlays* [3, 10, 31], also called Darknets, prevent such discovery by design. Mapping trust of individuals onto the system, they allow connections between devices only if their owners share a mutual trust relationship in real life. Social Overlays hence evolve topologies that reconstruct subgraphs of the social network of their participating individuals. Both explicitly (e.g. the profile and messages) as well as implicitly shared data (e.g. participation and communication patterns) consequently can be hidden from untrusted parties.

Social Overlays currently are far from efficient enough to provide acceptable social networking or similar real-time communication services, and there is a distinct lack of analytical understanding that prevents significant enhancements. For a larger acceptance of such privacy-preserving overlays, it is necessary to design efficient routing algorithms with guaranteed convergence. Existing models for deterministic polylog routing in small-world networks assume a base graph in form of a lattice, which is not given in Social Overlays.

Our contribution to this complex topic is 1) a framework for analyzing routing algorithms in the described scenario, 2) a provably polylog routing algorithm based only on information about direct neighbors and 3) an analysis of the gain achieved by additionally considering the two-hop neighborhood for routing decisions.

## 1.1 Social Overlays

Overlays in general are application layer networks. Formally, they are represented by a graph  $G = (V, E)$  of nodes  $V$  and edges  $E$  between nodes. Structured peer-to-peer systems, including distributed hash-tables (DHTs), introduce a metric namespace  $M$  and a function  $dist : M \times M \rightarrow \mathbb{R}$ , indicating the distance of two identifiers within this namespace. Each node  $u \in V$  is mapped to an identifier  $id(u) \in M$ . Edges are then chosen in such a way that the standard routing algorithm is guaranteed to converge in a polylog number of steps.

Social Overlays, limited by the constraint of establishing connections only between devices of individuals with mutual trust, are prevented from creating such topologies. Early approaches for Darknets, for example Turtle [25], use flooding, and hence are aimed at rather small group sizes. Probabilistic search has been implemented in OneSwarm [16], a Darknet protocol for BitTorrent. Both approaches can lead to large overhead, low success rates and long routes in case of rare files and sparse topologies. Second-level overlays have been proposed to decrease delays and overhead: MCON and XVine [24, 31] hence implement structured peer-to-peer systems by connecting the closest neighbors in the namespace through tunnels on the Social Overlay. Discovering and maintaining these tunnels under churn, however, introduces a high overhead. They furthermore are characterized by high delays, which make them unsuitable for most social applications. GUNet, an anonymous publication system with a Darknet mode, uses recursive Kademlia for routing, restricting the neighbors to trusted contacts [10]. Consequently, routing frequently terminates in dead ends, i.e. when a node is reached without any neighbor closer to the target. It hence requires a high replication rate to still locate content. All given approaches have mainly been proposed for anonymous file-sharing with a high replication rate for popular files. They are not designed to provide social networking or real-time communication services.

*Embedding* a routing structure into the social graph has been proposed as an alternative solution to increase the efficiency of Social Overlays. Formally, an embedding is a function  $F : V \rightarrow M$  from the set of nodes into a suitable metric namespace. Though any such function  $F$  qualifies as an embedding, the aim is to approximate a routing structure, which allows an algorithm to efficiently route messages from any source  $s \in V$  to any destination  $t \in V$  based on local knowledge. The ratio between the length of the routed paths compared to the length of the actual shortest paths within the overlay is commonly termed as the *stretch*. The ideal case would be a *no-stretch* embedding, however, no algorithm is known to achieve this, so that  $M$  requires

only a polylog number of dimensions. Embeddings that allow the standard routing algorithm to terminate successfully for all source-target pairs are called *greedy embeddings*. They achieve that nodes share edges with those closest to them in the namespace. In other words: An embedding  $F$  is called greedy if for all distinct node pairs  $s, t \in V$ ,  $s$  has a neighbor  $v$  that is closer to  $t$ , i.e.  $\text{dist}(F(v), F(t)) < \text{dist}(F(s), F(t))$ . Extensive research has been performed on *greedy embeddings* [7, 9, 11, 15, 18, 23, 32], especially for wireless sensor networks and Internet routing. All these approaches share the idea of constructing a spanning tree of the graph, which then is embedded into a hyperbolic, euclidean or custom-metric space. The resulting embedding is a greedy embedding of the complete graph as well. Routing along the tree is always successful, and shorter paths may be found using additional edges as short cuts. Dynamic node participation and potentially adversarial activities, however, require costly re-computation and maintenance of spanning tree and embedding. The central role of the root node additionally represents a perfect target for attacks, and the identifiers disclose the structure of the spanning tree.

Several more robust and privacy preserving embeddings have been proposed to meet Darknet requirements [8, 29, 30]. Rather than creating and embedding a spanning tree, these approaches aim at embedding the social graph in a  $m$ -dimensional lattice using periodic adjustments of the node identifiers. In the Darknet mode in Freenet, for example, the social graph is embedded in a ring over the namespace  $[0, 1)$ . All nodes choose identifiers from this namespace randomly when joining. Periodically, a node selects a partner sampled by a short random walk. The two nodes decide if they swap identifiers for an increased accuracy of the embedding, i.e. a better approximation of a ring structure over the topology. The resulting embeddings of these approaches are inaccurate. For instance, nodes that are neighbors in the namespace not necessarily are topological neighbors. The standard routing algorithm hence fails and has to be adapted to deal with local optima during the routing process.

Freenet [4] suggests a *distance-directed depth-first search* to mitigate inaccuracies in the embedding. Messages are forwarded to the neighbor closest to the destination that has not been contacted before. A backtracking phase starts if a node has no neighbors left to contact or repeatedly receives the same message. This algorithm has a low performance in large networks. The current implementation of Freenet hence additionally uses information about the identifiers of the two-hop neighborhood, thus implementing a *Neighbor-of-Neighbor* (NoN) routing algorithm.

## 1.2 Routing: Models, Algorithms, and Complexity

When analyzing decentralized routing algorithms, the most intensively studied property is the (maximal) expected routing length. Let  $V$  be the set of nodes and  $R^A(s, t)$  denote the number of steps needed to route from node  $s$  to node  $t$  using algorithm  $A$ . The maximal expected routing length is then given by  $\max_{s, t \in V} \mathbb{E}(R^A(s, t))$ . The expected routing length is similarly defined as  $\frac{1}{|V|(|V|-1)} \sum_{s \neq t \in V} \mathbb{E}(R^A(s, t))$ .

One of the first models for the analysis of routing in small-world graphs has been proposed by Kleinberg [17]. Here, nodes are placed on a  $m$ -dimensional lattice. Each node  $v$  then is connected to all nodes within distance  $p \geq 1$  and additionally has  $q \geq 1$  long-range contacts. A long-range contact  $u$  is chosen with probability anti-proportional to  $d^r$  for some  $r > 0$ , where  $d$  is the distance of  $v$  to  $u$ . The routing length of the standard algorithm with respect to the described topology model is polylog if and only if  $r = m$ . These results for the case  $r = m$  have been extended in various ways: It has been shown that the standard routing algorithm has expected routing length  $\Theta(\log^2 n)$  steps. Since the diameter is logarithmic, this is not asymptotically optimal. Consequently, extensions of the routing algorithm using the information of  $\lceil \log n \rceil$  nodes in each step have been proposed, which reduce the expected routing length to  $\Theta(\log^{1+1/m} n)$  [21]. Similar alternative routing algorithms, considering a larger neighborhood before choosing the next hop, have been discussed in [14, 19]. Though achieving close to optimal or optimal performance, these algorithms are designed considering a constant degree distribution. Furthermore, they are based on additional knowledge about the network size, which is not supposed to be known in a privacy-preserving embedding. Closer related to the topic of Social

Overlays, the standard routing algorithm has been analyzed in case the degree of a node is chosen according to a scale-free distribution with exponent  $\alpha$ . The expected routing length for directed scale-free graphs is asymptotically the same as in the original model, but in case of undirected links, it is reduced to  $\mathcal{O}(\log^{\alpha-1} n \log \log n)$  [12]. The case of using Neighbor-of-Neighbor (NoN) information for routing has been treated in [20], finding that with  $\Theta(\log n)$  neighbors per node the expected routing length is asymptotically equal to the diameter.

Additionally to the routing performance, various properties of small-world models have been analyzed. Detailed studies on the diameter of such graphs with regard to the clustering exponent  $r$  have been made [5, 22]. Furthermore, an generative model on how long-range links are created by a random process modeling the movement of individuals over time has been suggested [2]. However, all works assume an underlying lattice structure, so that each node shares an edge to those that are closest to it. Considering arbitrary base graphs rather than lattices leads to an expected routing length of  $n^{\Omega(1)}$  [13]. We assume that our embedding algorithms provide an enhanced structure and thus a lower routing complexity than using unstructured graphs. To the best of our knowledge, the only result about local edges is that they are necessary for the connectivity of the graph [21]. Hence, though heuristic embedding algorithms do not achieve links between nodes closest in the namespace, connectivity and routing success require some type of local connections. We build our models considering, extending and complementing the above results. Our main modification lies in introducing a parameter governing the accuracy of the embedding, while at the same time guaranteeing connectivity always surely.

### 1.3 Prior Work and Contributions

In prior work, we have extended Kleinberg's model to address the expected inaccuracy of heuristic embeddings. Nodes hence are not connected to their closest, but to nodes within a specified distance in a lattice [27]. The accuracy of the embedding is reflected by the maximal distance  $C$  between closest neighbors. We also have shown that the Freenet algorithm does not achieve polylog routing paths and suggested *NextBestOnce* [28].

In this paper, we prove that *NextBestOnce* has polylog maximal expected routing length for sufficiently accurate embeddings of social graphs. We model social graphs as graphs with a scale-free degree distribution with exponent  $\alpha$ . Additionally, we quantify the gain of using information about the neighbors' neighbors for routing. The extended algorithm *NextBestOnce-NoN* is shown to have a maximal expected routing length of  $\mathcal{O}(\log^{\delta(\alpha)(\alpha-1)} n \log \log n + C^3 \log n)$  for  $\delta(\alpha) < 1$ , whereas *NextBestOnce* only achieves an expected routing length of  $\Omega(\log^{(\alpha-1)} n + C)$  and  $\mathcal{O}(\log^{(\alpha-1)} n \log \log n + C^3 \log n)$ .

The existence of local minima with regard to the distance to the target complicates the analysis decisively. Our methodology needs to depart from the traditional analysis of routing as an integer-valued decreasing random process. Various techniques for probabilistically bounding the increase due to such a local minima have been exploited to provide the required proofs.

We start by precisely defining the model and introducing the routing algorithms in Section 2. Afterwards, we state our results with short sketches of the proofs in Section 3. In Sections 4 and 5 the proof for the upper and lower on *NextBestOnce* are presented. *NextBestOnce-NoN* is treated in Section 6. The paper is completed by a discussion of the results and their impact in Section 7.

## 2 Preliminaries

In this section, our graph model of a Social Overlay, routing algorithms, and central definitions are introduced.

## 2.1 Modelling inaccurate embeddings

We use a model for restricted topologies with heuristic embeddings, an extension to Kleinberg's small-world model [17]. Though Kleinberg's model proposes an explanation how short paths are found in small-world networks, it is only of restricted use with respect to Social Overlays. The main discrepancy between an embedded trust graph and Kleinberg's model is the underlying lattice structure of the latter. The currently employed privacy-preserving embedding algorithms cannot achieve a greedy embedding. Rather, they result in nodes that are not connected to those that are closest with regard to the distance of identifiers. Our Social Overlay model provides a parameter  $C$  for the accuracy of the embedding. Furthermore, Kleinberg's model is extended to allow for arbitrary degree distributions and undirected graphs [28].

A graph of the class  $\mathcal{D}(n, m, C, L)$  consists of  $n^m$  nodes, arranged in a  $m$ -dimensional hypercube of side length  $n$ , so nodes are given unique identifiers (IDs) in  $\mathbb{Z}_n^m$ . In the following, we use the name  $v$  of a node synonymously with its identifier  $id(v) = (v_1, \dots, v_m)$ . The distance between two nodes  $u$  and  $v$  is given by the Manhattan distance with wrap-around:

$$dist(v, u) = \sum_{i=1}^d \min\{|v_i - u_i|, n - |v_i - u_i|\}$$

The parameter  $C$  is a measure for the accuracy of the embedding, and gives the maximal distance to the closest neighbor in each principal direction.

More precisely, each node  $v = (v_1, \dots, v_m)$  is given short-range links to neighbors  $a_1^v, \dots, a_m^v, b_1^v, \dots, b_m^v$ . Here  $a_j^v$  is chosen from the set

$$A_j^v = \{u = (u_1, \dots, u_m) \in V : u_i = v_i \text{ for } i \neq j, 1 \leq \min\{u_j - v_j, n + u_j - v_j\} \leq C\}.$$

Analogously,  $b_j^v$  is chosen from

$$B_j^v = \{u = (u_1, \dots, u_m) \in V : u_i = v_i \text{ for } i \neq j, 1 \leq \min\{v_j - u_j, n + v_j - u_j\} \leq C\}.$$

The random variable  $L$  governs the degree distribution, an inherent property of the trust graph. In addition to the short-range links, long-range links are chosen in a two step process:

1. choose a label  $l_v \in \mathbb{N}$ , distributed according to  $L$ , for each node  $v \in V$
2. connect nodes  $u, v$  with probability

$$P(l(u, v) | l_u = l_1, l_v = l_2, dist(u, v) = d) = 1 - e^{-\frac{l_1 l_2}{d^m \gamma}} \quad (1)$$

where  $\gamma$  is a normalization constant chosen such that

$$2 \sum_{d=1}^{n/2} \sum_{l_1=1}^{\infty} \left(1 - e^{-\frac{l_1}{d^m \gamma}}\right) P(L = l_1) = 1, \quad (2)$$

i.e. the expected number of long-range links of a node  $v$  with label  $l_v = 1$  is 1.

Note that we abbreviate the event  $\{l(u, v) = 1\}$  by  $l(u, v)$ . In general, brackets indicating events are dropped to enhance readability in later sections. The above model has proven useful in analyzing routing alternative to a memoryless greedy approach, which is bound to fail in case the embedding is not greedy.

Fraigniaud and Giakkoupis [12] also analyzed routing in small-world networks with a scale-free degree distribution. In their generative model, long-range links are first created as directed edges and then the reverse edges are added. This approach complicates an analysis of NoN routing, since one has to distinguish in which direction edges were originally selected.

## 2.2 Routing Algorithms

For deterministic routing based on a non-greedy embedding, state information is needed to avoid loops and dead ends. Backtracking is used in case a node has no suitable neighbor to forward the message to. Furthermore, nodes are *marked* when they should only be contacted for backtracking in the future.

---

**Algorithm 1** NextBestOnce\*(Node  $v$ , Node  $p$ , ID  $t$ , Set  $B$ , boolean  $b$ )

---

```

1: # input:  $v$ : message holder,  $p$ : predecessor,  $t$ : target,  $B$ : marked nodes,  $b$ : backtracking?
2: #  $N_v$ : neighbors of  $v$ 
3: #  $IDS(u)$ : set of identifiers associated with  $u$ 
4: if  $id(v) == t$  then
5:   routing successful; terminate
6: end if
7: if  $\neg b$  then
8:    $v.predecessor.add(p)$ ;
9: end if
10:  $S = \{u \in N_v : \neg B.contains(u)\}$ 
11: if  $S$  NOT EMPTY then
12:    $nextNode = argmin_{u \in S} dist(IDS(u), t)$ 
13:    $b = \text{false}$ ;
14:   if  $dist(nextNode, t) \geq dist(v, t)$  then
15:      $B.add(v)$ 
16:   end if
17: else
18:    $B.add(v)$ 
19:    $nextNode = v.predecessor.pop()$ ;
20:    $b = \text{true}$ ;
21: end if
22: if  $nextNode \neq \text{null}$  then
23:   NextBestOnce( $v$ ,  $t$ ,  $nextNode$ ,  $B$ ,  $b$ )
24: else
25:   routing failed; terminate
26: end if

```

---

The order by which nodes are *marked* is crucial for the routing performance. The straight-forward approach, currently implemented in Freenet, is a distance-directed depth-first search, *marking* nodes the first time they are contacted. However, this algorithm does not achieve polylog expected routing length, as is shown in [28]. Consequently, *NextBestOnce* was introduced, which allows a node to contact all neighbors closer to the destination before *marking* it. *NextBestOnce* has been shown to achieve polylog maximal expected routing length for constant  $C$  for a simplified version of the above model, but simulations indicate that for realistic network sizes, the performance gain in comparison to Freenet routing is limited [27, 28].

Hence, we suggest to enhance the performance of the algorithm by using additional information. Rather than only one identifier, each node provides a set of identifiers. The extended algorithm, *NextBestOnce\**, is described in Algorithm 1. The input of *NextBestOnce\** consists of the current message holder  $v$ , the predecessor  $p$  of  $v$ , the target ID  $t$ , the set  $B$  of *marked* nodes, and a flag  $b$  indicating if the routing is in the backtracking phase. Note that  $B$  can be realized in a privacy-preserving manner, e.g. by relying on a bloom filter, and is not decisive for the asymptotic routing length. Each node keeps a stack of predecessors for backtracking, which are contacted if  $v$  has only *marked* neighbors (ll. 19-20).

If at least one neighbor is not *marked*,  $v$  selects the not *marked* neighbor  $u$ , so that the distance to one of the identifiers in  $IDS(u)$  is minimal (l. 12). For *NextBestOnce*, this set of identifiers only consists of the ID of  $u$ , the generic version allows a node to provide multiple IDs. Though we focus on the case that  $IDS(u)$  consists of the IDs of  $u$  and its neighbors, the algorithm *NextBestOnce\** allows for e.g. multiple realities as well.

After determining the next node  $u$  on the path,  $v$  is *marked* if  $u$  is at a larger distance to  $t$  (l.

15) or backtracking starts (1.18). To guarantee termination, only one representative ID of  $u$  is considered for the decision of *marking*  $v$ .

### 2.3 Definitions and Notation

In the remainder of the paper, we analyze the performance of two routing algorithms based on *NextBestOnce*<sup>\*</sup>. The first one, *NextBestOnce* has been proposed in [27] and only uses the identifiers of the direct neighbors, i.e.  $IDs(u) = id(u)$  in Algorithm 1. The second algorithm, *NextBestOnce-NoN*, uses information about neighbors of neighbors, i.e.  $IDs(u) = \{id(u)\} \cup \bigcup_{v \in N(u)} \{id(v)\}$ , where  $N(u)$  is the set of neighbors of node  $u$ .

The number of hops required by *NextBestOnce*, respectively *NextBestOnce-NoN*, to find a path from source  $s$  to destination  $st$  is denoted by  $R^{NBO}(s, t)$  and  $R^{NoN}(s, t)$ , respectively.

The performance is analyzed with regard to the Social Overlay model presented above. Labels are chosen according to a scale-free distribution  $S_\alpha$  with exponent  $2 < \alpha < 3$  and a maximum  $\mu$ , i.e.

$$P(S_\alpha = k) \propto \frac{1}{k^\alpha}, \quad k = 1 \dots \mu. \quad (3)$$

Scale-free degree distribution are common in various complex networks, especially social networks. Furthermore, the set  $B_d(v) = \{u : dist(v, u) < d\}$  contains all nodes at distance less than  $d$  of  $v$ .

For reasons of presentation, results are given for  $m = 1$  dimensions, but can analogously be derived for multi-dimensional identifier spaces.

## 3 Results

We present upper and lower bounds for the performance of *NextBestOnce*. The routing length increases at least linearly with  $C$ , the maximal distance to a local neighbor. If  $C$  is constant, the bounds are the same as those in [12] for a small-world model with a scale-free degree distribution and edges between all nodes within distance 1. This agreement is non-trivial, since it has been shown in [28] that straight-forward extensions to the standard routing algorithm do not achieve polylog performance for  $C > 2$ .

**Theorem 3.1.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  and two nodes  $s, t \in V$  with distance  $d = dist(s, t)$ , an upper bound on the expected routing length of *NextBestOnce* is given by*

$$\mathbb{E}(R^{NBO}(s, t)) = \mathcal{O}(\log^{\alpha-1} d \log \log d + C^3 \log n). \quad (4)$$

*The maximal expected routing length is consequently*

$$\max_{s, t \in V} \mathbb{E}(R^{NBO}(s, t)) = \mathcal{O}(\log^{\alpha-1} n \log \log n + C^3 \log n). \quad (5)$$

For the proof two routing phases are considered. First, the number of steps to reach a node within distance  $C$  of the target is bound by  $\mathcal{O}(\log^{\alpha-1} n \log \log n)$ . Note that up to this point, the distance to the target can be modeled as a monotonously decreasing random process. Hence, the proof is essentially the same as in [12]. Our contribution lies in bounding the remaining number of steps. This consists of a) showing that with high probability no node at a distance exceeding  $C^2 \log n$  is contacted from this point on, and b) the worst case complexity of *NextBestOnce* on a graph of size  $N$  is  $\mathcal{O}(CN)$ . The bound then follows from applying b) to the subgraph of size  $C^2 \log n$ . The proof is presented in Section 4.

**Theorem 3.2.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  with  $C < \frac{1}{4}n^{1/4}$  and two nodes  $s, t \in V$ , a lower bound on the expected routing length of *NextBestOnce* is given by*

$$\frac{1}{n(n-1)} \sum_{s \neq t \in V} \mathbb{E}(R^{NBO}(s, t)) = \Omega(\log^{\alpha-1} n + C) \quad (6)$$

Again, the first term follows essentially from [12], bounding the number of steps to reach a node within distance  $C$  of the target. The second term is derived by considering that with constant probability  $t$  and its only 2 local neighbors have at most one long-range link each. Conditioning on this event, it can be shown that again with constant probability routing needs at least  $C/4$  steps after reaching a node within distance  $C$  of  $t$ . The proof can be found in Section 5.

This concludes our results for *NextBestOnce*. The upper bound guarantees polylog routing length as long as  $C$  is polylog. The lower bound provides a way to measure the gain of NoN information for routing. We show that when considering NoN information the expected routing length is reduced by more than a constant factor, although the average degree is constant. Indeed, the routing length  $T$  of *NextBestOnce* is reduced to  $T^\delta$  for some  $0 < \delta < 1$  in case of *NextBestOnce-NoN*.

**Theorem 3.3.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$ , an upper bound on the maximal expected routing length of NextBestOnce-NoN is given by*

$$\max_{s, t \in V} \mathbb{E}(R^{NoN}(s, t)) = \mathcal{O} \left( \log^{\delta(\alpha)(\alpha-1)} n \log \log n + C^3 \log n \right) \text{ for } \delta(\alpha) = 1 - \frac{(\alpha-2)(3-\alpha)}{\alpha}. \quad (7)$$

The proof is rather lengthy. Selecting parameters  $0 \leq k \leq \alpha - 2$ , and  $1/2 \leq r \leq 1$ , we bound the number of steps to reach a node within distance  $e^{\log^r n}$  of  $t$  by  $\mathcal{O} \left( \log^{\alpha-r-k(3-\alpha)} n \log \log n \right)$ . The idea is to determine the probability of halving the distance in the next two steps. For this purpose, the probability of contacting a nodes of degree  $\log^k n$  and  $\log n$  is derived. With constant probability, the later has a neighbor at half its distance to  $t$ . From this, the above bound for the first phase can then be derived using basic results about stochastic processes. The steps needed to cover the remaining distance are then at most  $\mathcal{O} \left( \log^{r(\alpha-1)} n + C^3 \log n \right)$  by Theorem 3.1. Afterwards, the result is obtained by finding the minimum of a two-dimensional extremal value problem with variables  $k$  and  $r$ . The complete proof is presented in Section 6.

Our results are obtained under the assumption that the graph  $G$  is connected. This holds for  $C = \mathcal{O}(n^s)$  for any  $s < 1$  with overwhelming probability. The proof of this follows easily from Lemma 4.2 and can be found in [26].

## 4 Proof of Theorem 3.1

The upper bound on *NextBestOnce*'s routing length is derived by dividing the routing into two phases: the number of steps  $R_1^{NBO}(s, t)$  needed to reach a node  $v$  within distance  $C$  of  $t$  and the number of steps  $R_2^{NBO}(s, t)$  to reach  $t$  from  $v$ .

**Lemma 4.1.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  and two nodes  $s, t \in V$  with  $d = \text{dist}(s, t)$ , the expected routing length of NextBestOnce during the first phase is*

$$\mathbb{E}(R_1^{NBO}(s, t)) = \mathcal{O}(\log^{\alpha-1} d \log \log d)$$

Since the distance to  $t$  decreases by at least 1 in each step during the first phase, the above lemma is essentially treated in [12]. A complete proof for our slightly different model can be found in [26].

In the following, we prove that the remaining distance is covered in  $\mathcal{O}(C^3 \log n)$  steps. This requires two preliminary results: It needs to be shown that there exist polylog paths between two nodes within distance  $C$ , and that these paths are found by *NextBestOnce*. Lemma 4.2 gives the probability that two nodes are connected by a so called *greedy* path, i.e. a path  $u_0, u_1, \dots, u_{l+1}$ , so that  $\text{dist}(u_i, v) < \text{dist}(u_{i-1}, v)$  for  $i = 0, \dots, l$ . Let  $g(v, w)$  indicate if  $v$  and  $w$  are connected by a greedy path. For brevity, we write  $P(g(v, w))$  rather than  $P(g(v, w) = 1)$ . Secondly, we show in Lemma 4.3 that *NextBestOnce* has routing length  $\mathcal{O}(CN)$  on any (sub-)graph of order  $N$ . Finally, Lemma 4.2 is applied to show that with overwhelming probability no node at distance  $\Omega(C^2 \log n)$  to  $t$  is contacted, so that the bound  $\mathcal{O}(C^3 \log n)$  is a consequence of Lemma 4.3.



**Lemma 4.2.** For two nodes  $w, v \in V$  with  $\text{dist}(w, v) > C^2 \log n$ , the probability that  $w, v$  are connected by a greedy path is  $P(g(w, v)) = \Omega\left(1 - \frac{1}{n}\right)$

*Proof.* Recall from Section 2.1 that each node  $u$  has two short-range neighbors  $a_1^u$  and  $b_1^u$  chosen independently of each other. They are both within distance  $C$  of  $u$ , but in opposite directions.

For all pairs  $(v, w) \in V \times V$ , there is a path of short-range links of length at most  $C$  originating at  $v$  leading to a node within distance  $C$  of  $w$  and via versa. A greedy path between  $v$  and  $w$  exists if those two paths intersect (see Figure 1). Denote by  $g^a(u_0, u_{l+1})$  the event that  $u_0$  and  $u_{l+1}$  are connected by a greedy path and  $a_1^{u_i} = u_{i+1}$  for  $i = 0, \dots, l$ .  $g^b(u_0, u_{l+1})$  is defined analogously. Without loss of generality,  $w$  is 'above'  $v$  in the namespace, i.e.  $\text{dist}(w, v) = w - v \bmod n$ .

In the following, we show that  $P(g^a(v, u)) \geq \frac{1}{C}$  for all  $u \in U := \{u \in V : \text{dist}(v, w) = \text{dist}(v, u) + \text{dist}(u, w)\}$ , i.e. all nodes in the shorter ring segment between  $v$  and  $w$ . If  $\text{dist}(v, u) \leq C$ , then  $P(g^a(v, u)) \geq P(a_1^v = u) = \frac{1}{C}$  by the choice of short-range links. Otherwise, there exists a node  $z$  such that  $g^a(v, z)$  holds and  $\text{dist}(v, u) - C \leq \text{dist}(v, z) < \text{dist}(v, u)$ . It follows that  $P(g^a(v, u)) \geq P(a_1^z = u) = \frac{1}{C}$ . Similarly,  $P(g^b(w, u)) \geq \frac{1}{C}$  holds for all  $u \in U$ .

Because  $a_1^u$  and  $b_1^u$  are chosen independently, the probability that the two paths intersect can be bounded as follows:

$$\begin{aligned} P(g(v, w)) &\geq P\left(\bigcup_{u \in U} (g^a(v, u) \cap g^b(w, u))\right) \\ &\geq 1 - \left(1 - \frac{1}{C^2}\right)^{C^2 \log n} \\ &\geq 1 - e^{-\frac{C^2 \log n}{C^2}} \\ &= 1 - \frac{1}{n} \end{aligned}$$

The third inequality follows from  $1 - x \leq e^{-x}$  for  $x \in [0, 1]$ . □

As a second step, a worst-case bound on the routing length of *NextBestOnce* is needed.

**Lemma 4.3.** Let  $G = (V, E)$  be an undirected graph that is embedded in  $\mathbb{Z}_{|V|}$ , so that all  $v \in V$  are connected to nodes within distance  $C$  in each direction. The expected routing length of *NextBestOnce* on  $G$  is bounded by

$$\max_{s, t \in V} \mathbb{E}(R^{NBO}(s, t)) = \mathcal{O}(C|V|)$$

*Proof.* The algorithm definitively terminates after every node has been *marked*. We show that in average at least every  $C$ -th node is *marked*. The bound follows immediately. First note that the maximal increase in distance per hop is  $C$ : Each node  $u$  has a short-range link to a node  $v$ , so that the  $\text{dist}(u, t) < \text{dist}(v, t) \leq \text{dist}(u, t) + C$ .  $v$  is not yet *marked*, because a node is only *marked* after all neighbors closer to the destination, including the current message holder  $u$ , have been *marked*. Therefore, *NextBestOnce* can always choose a successor within distance  $C$ . The maximal path length without producing a circle is  $|V|$ . Assume the algorithm produces a circle of length  $l$ . Then at least  $l/C$  nodes on the circle are *marked*. To see this, recall that an increase in the distance  $t$  implies that a node is declared *marked*. In case of a circle the sum of the distance changes per hop equals zero, so the distance is increased in at least  $\frac{1}{C} = \frac{\min \text{Decrease}}{\max \text{Increase}}$  of all hops of the circle. The maximal number of hops without circles and the maximum number of hops in circles until all nodes have been *marked* give the bound

$$\max_{s, t \in V} \mathbb{E}(R^{NBO}(s, t)) \leq |V| + C|V| = \mathcal{O}(C|V|).$$

□

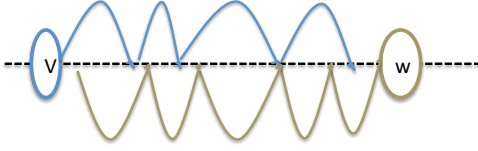


Figure 1: Path of edges of length at most  $C$  originating from  $v$ , respectively  $w$ . A greedy path between the  $v$  and  $w$  exists, because the two paths intersect.

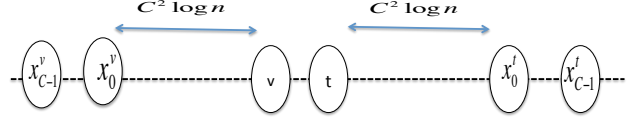


Figure 2: Nodes  $v$  and  $t$  and the corresponding set  $X$ : No node at a distance exceeding  $C^2 \log n + C$  is contacted with high probability.

It follows that the maximal number of steps is linear in the network size if the maximal increase to the destination is restricted by a parameter  $C$  independent of  $n$ . For arbitrary graphs, the algorithm terminates after  $\mathcal{O}(n^2)$  steps by Lemma 4.3. The last two lemmata enable us to bound the complexity of *NextBestOnce* during the second phase.

**Lemma 4.4.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  and two nodes  $s, t \in V$ , the expected routing length of *NextBestOnce* in the second phase is bounded by*

$$\mathbb{E}(R_2^{NBO}(s, t)) = \mathcal{O}(C^3 \log n).$$

*Proof.* Denote the first node in  $B_C(t)$  that is on the routing path by  $v$ . Consider the event  $A$  that no node at a distance exceeding  $C^2 \log n + C$  to the set  $\{v, t\}$  is contacted during the second phase of the routing. Using  $A^\perp$  for the complement of  $A$ , we get

$$\begin{aligned} \mathbb{E}(R_2^{NBO}(s, t)) &= P(A) \cdot \mathbb{E}(R_2^{NBO}(s, t) | A) + (1 - P(A)) \cdot \mathbb{E}(R_2^{NBO}(s, t) | A^\perp) \\ &= P(A) \cdot \mathcal{O}(C^3 \log n) + (1 - P(A)) \cdot \mathcal{O}(Cn) \end{aligned}$$

The last step follows from applying Lemma 4.3 to the subgraph of size  $C^2 \log n$  as well as to the whole graph  $G$ .

It remains to determine  $P(A)$ . The claim holds if  $v = t$ . Otherwise, let  $x_i^v$  for  $i = 0, \dots, C - 1$  be the node such that  $\text{dist}(x_i^v, t) = C^2 \log n + i$  and  $\text{dist}(x_i^v, t) > \text{dist}(x_i^v, v)$ . Analogously,  $x_i^t$  denotes the node such that  $\text{dist}(x_i^t, v) = C^2 \log n + i$  and  $\text{dist}(x_i^t, v) > \text{dist}(x_i^t, t)$ . The set  $X = \{x_i^u : u \in \{v, t\}, 0 \leq i < C\}$  consists of two sets of consecutive nodes at distance  $C^2 \log n$  to  $C^2 \log n + C - 1$  from the set  $\{v, t\}$  (see Figure 2). If a node at a higher distance than  $C^2 \log n$  is reached after  $v$ , at least one node in  $X$  needs to be on the path as well, because the maximal regression per hop is bound by  $C$ . Recall that *NextBestOnce* marks a node  $u$  if all its neighbors closer to  $t$  have been marked. It follows recursively that if a successor at a higher distance than the current node  $u$  is chosen, all nodes reachable from  $u$  by paths along which the distance to  $t$  decreases monotonously have been marked. Consequently, a node  $u$  with  $\text{dist}(u, t) \geq C^2 \log n + C$  can only be on the path if all nodes  $x \in X$  do not have a greedy path to  $t$ .

Lemma 4.2 is applied to bound  $P(A)$  by the probability that all  $2C$  nodes in  $X$  have a greedy path to  $t$ :

$$P(A) \geq P\left(\bigcap_{x \in X} g(t, x)\right) = \Omega\left(\left(1 - \frac{1}{n}\right)^{2C}\right) = \Omega\left(1 - \frac{2C}{n}\right)$$

The last step holds since  $(1 - x)^k \geq 1 - kx$  for  $0 < x < 1$  and  $k > 1$ . Finally, we get

$$\mathbb{E}(R_2^{NBO}(s, t)) = \Omega\left(1 - \frac{2C}{n}\right) \cdot \mathcal{O}(C^3 \log n) + \mathcal{O}\left(\frac{2C}{n}\right) \cdot \mathcal{O}(Cn) = \mathcal{O}(C^3 \log n)$$

□

Theorem 3.1 is a direct consequence.

*Proof.* For a source-destination pair  $(s, t)$  with distance  $d = \text{dist}(s, t)$ , the expected routing length of *NextBestOnce* is bound by

$$\mathbb{E}(R^{NBO}(s, t)) = \mathbb{E}(R_1^{NBO}(s, t)) + \mathbb{E}(R_2^{NBO}(s, t)) = \mathcal{O}(\log^{\alpha-1} d \log \log d + C^3 \log n)$$

by Lemma 4.1 and 4.4. The distance between two nodes is at most  $n/2$ , so

$$\max_{s, t \in V} \mathbb{E}(R^{NBO}(s, t)) = \mathcal{O}(\log^{\alpha-1} n \log \log n + C^3 \log n).$$

as claimed.  $\square$

## 5 Proof of Theorem 3.2

As for the upper bound, the proof is done by dividing the routing process into two phases. Let  $R_1^{NBO}(s, t)$  be the number of nodes contacted to reach a node within distance  $C$  and  $R_2^{NBO}(s, t)$  the number of steps needed to get from this node to  $t$ .

**Lemma 5.1.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  and two nodes  $s, t \in V$ , the expected routing length for the first phase is*

$$\mathbb{E}(R_1^{NBO}(s, t)) = \Omega(\log^{\alpha-1} n).$$

The proof of Lemma 5.1 is very similar to the one presented in [12] and can be found in [26].

In order to show the second result, some facts about  $\mathcal{D}(n, 1, C, S_\alpha)$  are needed. Recall that  $l(u, v)$  denotes the event that there is a long-range link incident to  $v$  and  $u$ .

**Lemma 5.2.** *The probability that a long-range link is at least of length  $2\sqrt{n}$  is constant, i.e.*

$$P(\text{dist}(u, v) \geq 2\sqrt{n} | l(u, v)) = \Omega(1).$$

*Proof.* First, consider that the probability that  $u$  and  $v$  is given by the following (where  $\gamma = \Theta(\log n)$  is the normalization constant in Section 2):

$$\begin{aligned} P(l(u, v) | \text{dist}(u, v) = d) &= \sum_{l_1=1}^{\infty} \sum_{l_2=1}^{\infty} \left(1 - e^{-\frac{l_1 l_2}{d\gamma}}\right) P(S_\alpha = l_1) P(S_\alpha = l_2) \\ &= \sum_{l_1=1}^{\infty} \sum_{l_2=1}^{\infty} \Theta\left(\frac{l_1 l_2}{d\gamma} P(S_\alpha = l_1) P(S_\alpha = l_2)\right) = \Theta\left(\frac{1}{\gamma d} \mathbb{E}(S_\alpha)^2\right) = \Theta\left(\frac{1}{\gamma d}\right). \end{aligned}$$

The last step holds since the expectation of  $S_\alpha$  is constant. Note that the probability that two randomly selected nodes on a ring of length  $n$  have at least distance  $\sqrt{n}$  converges to 1. The claim now easily follows:

$$\begin{aligned} P(\text{dist}(u, v) \geq 2\sqrt{n} | l(u, v)) &= \frac{P(l(u, v) | \text{dist}(u, v) \geq 2\sqrt{n})}{P(l(u, v))} P(\text{dist}(u, v) \geq 2\sqrt{n}) \\ &= \Omega\left(\frac{P(l(u, v) | \text{dist}(u, v) \geq 2\sqrt{n})}{P(l(u, v))}\right) = \Omega\left(\frac{\sum_{d=2\sqrt{n}}^{n/2} \frac{1}{\gamma d} \cdot \frac{2}{n}}{\sum_{d=1}^{n/2} \frac{1}{\gamma d} \cdot \frac{2}{n}}\right) = \Omega\left(\frac{\sum_{d=2\sqrt{n}}^{n/2} \frac{1}{d}}{\sum_{d=1}^{n/2} \frac{1}{d}}\right) \\ &= \Omega\left(\frac{\log(n/2) - \log(2\sqrt{n})}{\log(n/2)}\right) = \Omega\left(\frac{1/2 \log n - 4}{\log n}\right) = \Omega(1) \end{aligned}$$

The second last second step follows from  $\sum_{i=1}^n \frac{1}{i} = \Theta(\log n)$ .  $\square$

**Lemma 5.3.** *The expected number of nodes  $Q$  in  $V \setminus B_{\sqrt{n}}(t)$  that have a neighbor in  $B_d(t)$  for any  $d < \sqrt{n}$  is*

$$E(Q) = \Omega(d).$$

*Proof.* The claim follows from the fact that  $P(l(u, v) | \text{dist}(u, v) = d) = \Theta\left(\frac{1}{d \log n}\right)$  for any pair of nodes  $(u, v)$ .

$$\begin{aligned} E(Q) &= \sum_{d_1=\sqrt{n}}^{n/2} \sum_{d_2=0}^d \Theta\left(\frac{1}{(d_1 - d_2) \log n}\right) + \sum_{d_1=\sqrt{n}}^{n/2} \sum_{d_2=1}^d \Theta\left(\frac{1}{(d_1 + d_2) \log n}\right) \\ &= \Omega\left(\sum_{d_1=\sqrt{n}}^{n/2} \sum_{d_2=1}^d \frac{2}{2d_1 \log n}\right) = \Omega\left(\sum_{d_1=\sqrt{n}}^{n/2} \frac{2d}{2d_1 \log n}\right) = \Omega(d) \end{aligned}$$

The last step uses  $\sum_{d_1=\sqrt{n}}^{n/2} \frac{1}{d_1 \log n} = \Omega(1)$  as shown in the proof of Lemma 5.2.  $\square$

We can now derive a lower bound on  $R_2^{NBO}$ .

**Lemma 5.4.** For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  with  $C < \frac{1}{4}n^{1/4}$  and two nodes  $s, t \in V$ , the expected routing length for the second phase is

$$\mathbb{E}(R_2^{NBO}(s, t)) = \Omega(C).$$

*Proof.* The above bound is obtained by showing that  $P(R_2^{NBO}(s, t) \geq C/4 | A) = \Omega(1)$  for a suitable event  $A$  with  $P(A) = \Omega(1)$ . It follows directly that  $\mathbb{E}(R_2^{NBO}(s, t))$  grows at least linearly with  $C$ . Recall that  $t$  has two local neighbors  $a_1^t$  and  $b_1^t$  within distance  $C$  of  $t$ . The set of short-range neighbors of a node  $u$  is denoted by  $SN(u)$ , whereas  $LN(u)$  is the set of long-range neighbors. Furthermore, we abbreviate  $U = LN(t) \cup LN(a_1^t) \cup LN(b_1^t)$ . The event  $A = A_1 \cap A_2 \cap A_3 \cap A_4$  is the intersection of the following events:

- $A_1 = \{v \notin \{t, a_1^t, b_1^t\}\}$ : the first node within distance  $C$  of  $t$  is not  $t$ ,  $a_1^t$  or  $b_1^t$
- $A_2 = \{SN(t) = 2\}$ :  $a_1^t$  and  $b_1^t$  are  $t$ 's only short-range neighbors
- $A_3 = \{|LN(t)| \leq 1\} \cap \{|LN(a_1^t)| \leq 1\} \cap \{|LN(b_1^t)| \leq 1\}$ :  $t$  as well as its two short-range neighbors have maximally one long-range neighbor
- $A_4 = \bigcup_{u \in U} \{\text{dist}(u, t) \geq \sqrt{n}\}$ :  $t$  as well as its short-range neighbors have only long-range neighbors at distance at least  $\sqrt{n}$  to  $t$

Before showing that  $P(A) = \Omega(1)$ , note that indeed  $P(R_2^{NBO}(s, t) \geq C/4 | A) = \Omega(1)$ . *NextBestOnce* increases the distance to  $t$  by at most  $C$  in each step, hence by conditioning on  $A_3$  (and recalling that  $C \cdot C/4 < \sqrt{n}$ ),  $t$ ,  $a_1^t$  and  $b_1^t$  can not be contacted by a long-range neighbor in less than  $C/4$  steps. Therefore,  $t$  can only be found in less than  $C/4$  steps if a node on the path contacts either  $a_1^t$  or  $b_1^t$  via a short-range link (by event  $A_2$  and  $A_3$ ). The probability that a node  $u \in \{a_1^t, b_1^t\}$  is a short-range neighbor of a node  $w$  on the routing path  $X$  is

$$\begin{aligned} P(u \in SN(w) | w \in X) &= P(u \in \{a_1^w, b_1^w\} \cup w \in \{a_1^u, b_1^u\} | w \in X) \\ &\leq 2P(u \in \{a_1^w, b_1^w\} | w \in X) \leq 2P(u \in \{a_1^w, b_1^w\} | \text{dist}(u, w) \leq C) = \frac{2}{C}. \end{aligned} \tag{8}$$

The second last step holds because the probability that two nodes are short-range neighbors is maximal when their distance is at most  $C$ .

Applying an union bound, the probability that one of the first  $C/4$  nodes on the path after reaching  $v$  has an edge to either  $a_1^t$  or  $b_1^t$  is bounded by:

$$P(R_2^{NBO}(s, t) \geq C/4 | A) = \Omega\left(\left(1 - \frac{4}{C}\right)^{C/4}\right) = \Omega(1) \tag{9}$$

The last step holds, because  $(1 - 1/x)^x$  converges to  $1/e$  for  $x \rightarrow \infty$ .

It remains to show  $P(A) = \Omega(1)$ . Using independence of edge selection, we can rewrite:

$$P(A) = P(A_1 \cap A_2 \cap A_3 \cap A_4) = P(A_1 | A_2 \cap A_3 \cap A_4) P(A_2) P(A_3) P(A_4 | A_3)$$

For determining  $P(A_1|A_2 \cap A_3 \cap A_4)$ , we first define  $W_L^d = \{w \in V \setminus B_d(t) : l(w, B_C(t))\}$ , the set of all nodes with long-range links to a node at distance at least  $d$  that have links into  $B_C(t)$ . Similarly, let  $W_S = \{w \in V \setminus B_C(t) : SN(w) \cap B_C(t) \neq \emptyset\}$  be the set of nodes with short-range links into  $B_C(t)$ . Denote the predecessor of  $v$  on the routing path by  $w$ . We consider the complement of  $A_1$  to derive the desired bound.

$$\begin{aligned}
P(A_1^\perp | A_2 \cap A_3 \cap A_4) &\leq P(A_1^\perp \cap w \in W_L^C | A_2 \cap A_3 \cap A_4) + P(A_1^\perp \cap w \in W_S | A_2 \cap A_3 \cap A_4) \\
&\leq P(A_1^\perp \cap w \in W_L^{\sqrt{n}} | A_2 \cap A_3 \cap A_4) + P(A_1^\perp \cap w \in W_S | A_2 \cap A_3 \cap A_4) \\
&= P(A_1^\perp | A_2 \cap A_3 \cap A_4 \cap w \in W_L^{\sqrt{n}}) P(w \in W_L^{\sqrt{n}} | A_2 \cap A_3 \cap A_4) \\
&\quad + P(A_1^\perp | A_2 \cap A_3 \cap A_4 \cap w \in W_S) P(w \in W_S | A_2 \cap A_3 \cap A_4) \\
&\leq P(A_1^\perp | A_2 \cap A_3 \cap A_4 \cap w \in W_L^{\sqrt{n}}) + P(A_1^\perp | A_2 \cap A_3 \cap A_4 \cap w \in W_S) \\
&= \mathcal{O}\left(\frac{3}{C}\right) + \mathcal{O}\left(\frac{2}{C}\right)
\end{aligned} \tag{10}$$

If  $w \in W_L^{\sqrt{n}}$ , the short-range links of  $t$  do not influence  $A_1$ . For this reason, we can drop the condition  $A_2$  for the first term in Eq. 10. By Lemma 5.3 there are  $\Omega(C)$  nodes in  $V \setminus B_{\sqrt{n}(t)}$  having edges into  $B_C(t)$ . Conditioning on  $A_3$  and  $A_4$ , at most three of these long-range links are incidents to  $t$ ,  $a_1^t$  and  $b_1^t$ . The second summand  $\frac{2}{C}$  in Eq. 10 is derived as in Eq. 8. Note that  $A_3$  and  $A_4$  do not influence the event, given that  $w \in W_S$ . Consequently,

$$P(A_1 | A_2 \cap A_3 \cap A_4) = 1 - \mathcal{O}\left(\frac{3}{C} + \frac{2}{C}\right) = \Omega\left(1 - \frac{5}{C}\right) = \Omega(1)$$

$P(A_2)$  corresponds to the probability that none of the  $2(C-1)$  potential short-range neighbors but  $a_1^t, b_1^t$  have chosen  $t$  as a neighbor. So

$$P(A_2) = \left(1 - \frac{1}{C}\right)^{2(C-1)} = \Omega(1).$$

Similarly to Eq. 9, the last bound follows from  $(1 - 1/x)^{2x} \rightarrow e^{-2}$ . Long-range edges are selected independently, hence

$$\begin{aligned}
P(A_3) &= P(|LN(t)| \leq 1)^3 = \Omega\left((P(|LN(t)| \leq 1 | l_t = 1) P(l_t = 1))^3\right) \\
&= \Omega\left((P(|LN(t)| \leq 1 | l_t = 1) P(S_\alpha = 1))^3\right) = \Omega\left(P(|LN(t)| \leq 1 | l_t = 1)^3\right) = \Omega(1).
\end{aligned}$$

The second last step holds since  $P(S_\alpha = k) \propto k^\alpha$ . Furthermore, the last steps follows from Eq. 2, because an expected degree of 1 implies that the probability of having a degree of at most 1 is at least  $1/2$ .

For calculating  $P(A_4|A_3)$ , denote the long-range neighbor of  $t$ ,  $a_1^t, b_1^t$  by  $t_l, a_l$  and  $b_l$ , respectively. Because  $C < \sqrt{n}$ , it follows from  $\text{dist}(a, a_l) > 2\sqrt{n}$  that  $\text{dist}(t, a_l) > \sqrt{n}$ .

$$\begin{aligned}
P(A_4|A_3) &= P(\text{dist}(t_l, t) > \sqrt{n} | l(t_l, t)) \cdot P(\text{dist}(a_l, t) > \sqrt{n} | l(a_l, a_1^t)) \cdot P(\text{dist}(b_l, t) > \sqrt{n} | l(b_l, b_1^t)) \\
&\geq P(\text{dist}(t_l, t) > 2\sqrt{n} | l(t_l, t)) \cdot P(\text{dist}(a_l, a_1^t) > 2\sqrt{n} | l(a_l, a_1^t)) \cdot P(\text{dist}(b_l, t) > 2\sqrt{n} | l(b_l, b_1^t)) \\
&= P(\text{dist}(t_l, t) > 2\sqrt{n} | l(t_l, t))^3
\end{aligned}$$

Now  $P(A_4|A_3) = \Omega(1)$  is a direct consequence from Lemma 5.2. The above results confirm that indeed

$$P(A) = P(A_1|A_2 \cap A_3 \cap A_4)P(A_2)P(A_3)P(A_4|A_3) = \Omega(1 \cdot 1 \cdot 1) = \Omega(1).$$

Thus, we have shown that

$$P(R_2^{NBO}(s, t) \geq C/4) \geq P(R_2^{NBO}(s, t) \geq C/4|A) P(A) = \Omega(1).$$

Consequently, the expectation grows at least linearly in  $C$ , i.e.

$$\mathbb{E}(R_2^{NBO}(s, t)) = \Omega(C).$$

□

Theorem 3.2 follows from Lemma 5.1 and 5.4, because

$$\mathbb{E}(R^{NBO}(s, t)) = \mathbb{E}(R_1(s, t)) + \mathbb{E}(R_2(s, t)) = \Omega(\log^{\alpha-1} n + C).$$

## 6 Proof of Theorem 3.3

Fix  $1/2 \leq r \leq 1$  and  $0 \leq k \leq \alpha - 2$ . The routing is now split in two phases:  $R_1^{NoN}(s, t)$  gives the number of steps needed to get within distance  $e^{\log^r n}$  of  $t$ .  $R_2^{NoN}(s, t)$  is the number of steps to cover the remaining distance. For the proof, we assume that the maximum value of  $S_\alpha$  is  $\mu = \Theta(\log n)$ . Restricting the degree is obviously a relaxation, which avoids further case distinctions. The result holds for an unbounded maximum degree as well, as presented in [26]. We show that

$$\begin{aligned} \mathbb{E}(R^{NoN}(s, t)) &= \mathbb{E}(R_1^{NoN}(s, t)) + \mathbb{E}(R_2^{NoN}(s, t)) \\ &= \mathcal{O}\left(\log^{\alpha-r-k(3-\alpha)} n \log \log n\right) + \mathcal{O}\left(\log^{r(\alpha-1)} n \log \log n + C^3 \log n\right). \end{aligned}$$

The result is then obtained by finding  $r_{min}$  and  $k_{min}$  to minimize the above bound.

The bound for the second phase can be derived from the routing length of *NextBestOnce*.

**Lemma 6.1.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$ , two nodes  $s, t \in V$ , and  $1/2 \leq r \leq 1$ , the expected routing length of *NextBestOnce-NoN* after reaching a node within distance  $e^{\log^r n}$  of  $t$  is*

$$\mathbb{E}(R_2^{NoN}(s, t)) = \mathcal{O}\left(\log^{r(\alpha-1)} n \log \log n + C^3 \log n\right).$$

*Proof.* *NextBestOnce-NoN* is in expectation at least as fast as *NextBestOnce*, using the same procedure, only with additional information. Let  $u$  be the first node on the routing path with  $\text{dist}(t, u) \leq e^{\log^r n}$ . By Theorem 3.1, the expected routing length to get from  $u$  to  $t$  is:

$$\begin{aligned} \mathbb{E}(R_2^{NoN}(s, t)) &= \mathcal{O}(\mathbb{E}(R^{NBO}(u, t))) = \mathcal{O}\left(\log^{\alpha-1} e^{\log^r n} \log \log e^{\log^r n} + C^3 \log n\right) \\ &= \mathcal{O}\left(\log^{r(\alpha-1)} n \log \log n + C^3 \log n\right) \end{aligned}$$

This proves the claim. □

The first phase of the routing is considerable more work. In the following, assume  $C < e^{\log^{1/2} n}$ . Otherwise, the bound  $\mathcal{O}(C^3 \log n)$  holds for both *NextBestOnce* and *NextBestOnce-NoN* by Theorem 3.1. A preliminary Lemma is needed to determine the probability that nodes are adjacent given their labels.

**Lemma 6.2.** *Consider a node  $u$  with  $d = \text{dist}(u, t) > e^{\log^{1/2} n}$ , and a set  $W \subset V$ , so that  $\text{dist}(r, t) > d$  for all  $w \in W$ . Denote by  $V_{d'}^a = \{v \in V : v \in B_{d'}(t), l_v \geq a\}$  the set of all nodes within distance  $d' \leq d$  of the destination and label at least  $a$ . Furthermore, assume  $\frac{|W|}{(d-d') \log n} < 1/2$ . The probability that  $u$  is adjacent to a node in  $V_{d'}^a$ , conditioned on  $l_u$  and the absence of edges between  $W$  and  $B_{d'}(t)$ , is bounded by*

$$P\left(l(u, V_{d'}^a) | l_u = l \cap l(W, B_{d'}(t))^\perp\right) = \Omega\left(\frac{l}{\log n} [\log(d + d' - 1) - \log(d - d' + 1)] a^{2-\alpha}\right).$$

*Proof.* We show that the expected number  $Q$  of nodes  $v$  in  $V_{d'}^a$  that have a link to  $u$  is

$$E(Q) = \Omega \left( \frac{l}{\log n} [\log(d + d' - 1) - \log(d - d' + 1)] a^{2-\alpha} \right).$$

Then the probability of  $l(u, V_{d'}^a)$  satisfies this bound as well.

Proof of the last statement: For each  $v \in B_{d'}$ , the random variable  $Q_v$  is 1 if  $v \in V_{d'}^a$  and adjacent to  $u$ . Otherwise  $Q_v$  is 0. For the sum  $Q = \sum_{v \in B_{d'}} Q_v$ , it holds that

$$P(Q = 1) \geq 1 - e^{-E(Q)} = \Theta(E(Q)) \text{ for } E(Q) \rightarrow 0$$

The first inequality follows from the fact that the  $Q_v$  are independent and  $1 - x \leq e^{-x}$  for  $x \in [0, 1]$ :

$$P(Q = 0) = \prod_{i=1}^r P(Q_i = 0) = \prod_{i=1}^r (1 - E(Q_i)) \leq \prod_{i=1}^r e^{-E(Q_i)} = e^{-E(Q)}$$

In the following,  $E(Q)$  is computed as the sum of probabilities that each node belongs to the set of neighbors of  $u$  within  $V_{d'}^a$ . Furthermore, note that in case of a one-dimensional ID space (, i.e. a ring), a node with distance at most  $d' - 1$  to  $t$  has a distance between  $d - d' + 1$  and  $d + d' - 1$  to  $u$ .

$$\begin{aligned} E(Q) &= \sum_{i=d-d'+1}^{d+d'-1} P \left( l(u, v) \cap v \in V_{d'}^a | l_u = l \cap l(W, B_{d'}(t))^\perp \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i \right) \\ &= \sum_{i=d-d'+1}^{d+d'-1} \sum_{j=a}^{\infty} P \left( l(u, v) \cap l_v = j | l_u = l \cap l(W, B_{d'}(t))^\perp \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i \right) \\ &= \sum_{i=d-d'+1}^{d+d'-1} \sum_{j=a}^{\infty} P \left( l_v = j | l_u = l \cap l(W, B_{d'}(t))^\perp \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i \right) \\ &\quad \cdot P \left( l(u, v) | l_u = l \cap l(W, B_{d'}(t))^\perp \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i \cap l_v = j \right) \\ &:= \sum_{i=d-d'+1}^{d+d'-1} \sum_{j=a}^{\infty} t_1 \cdot t_2 \end{aligned} \tag{11}$$

Recall from Eq. 1 in Section 2 that two nodes  $v, w$  are adjacent with probability

$$P(l(v, w) | \text{dist}(v, w) = i \cap l_v = l_1 \cap l_w = l_2) = 1 - e^{-\frac{l_1 l_2}{i\gamma}} = \Theta \left( \frac{l_1 l_2}{i\gamma} \right) = \Theta \left( \frac{l_1 l_2}{i \log n} \right). \tag{12}$$

The last steps holds due to Eq. 2. A scale-free degree distribution  $S_\alpha$  as defined in Eq. 3 is used, i.e. the probability that a node  $v$  has label  $l$  is proportional to  $l^{-\alpha}$ . The probability that a node  $v \in B_{d'}$  is adjacent to a node in  $W$  is obtained by a simple union bound.

$$\begin{aligned} &P(l(v, W) | l_v = l \cap \text{dist}(t, W) > d \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) \\ &\leq \sum_{w \in W} P(l(v, w) | l_v = l \cap \text{dist}(t, W) > d \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) \\ &= \sum_{w \in W} P(l(v, w) | \text{dist}(v, w) > d - d' \cap l_v = l \cap \text{dist}(v, u) = i) \\ &= \Theta \left( \frac{l|W|}{(d - d') \log n} \right) \end{aligned}$$

The last step holds by Eq. 12. Because the expectation  $E(S_\alpha)$  is constant, we get:

$$\begin{aligned}
& P(l(v, W) | \text{dist}(t, W) > d \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) \\
&= \sum_{l=1}^{\infty} P(l(v, W) | l_v = l \cap \text{dist}(t, W) > d \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) P(l_v = l) \\
&= \Theta \left( \sum_{l=1}^{\infty} \frac{l|W|}{(d-d') \log n} P(l_v = l) \right) \\
&= \Theta \left( \frac{|W|}{(d-d') \log n} \right)
\end{aligned}$$

By assumption,  $\frac{|W|M}{(d-d') \log n} < 1/2$ , and hence both  $P(l(W, v)^\perp | v \in B_{d'}(t) \cap \text{dist}(v, u) = i) = \Omega(1)$  and  $P(l(W, v)^\perp | l_v = j \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) = \Omega(1)$ . Given that labels and edges are chosen independently,  $t_1$  is easily obtained as:

$$\begin{aligned}
t_1 &= P(l_v = j | l(W, v)^\perp \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) \\
&= \frac{P(l(W, v)^\perp | l_v = j \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) P(l_v = j \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i)}{P(l(W, v)^\perp | v \in B_{d'}(t)) P(v \in B_{d'}(t) \cap \text{dist}(v, u) = i)} \\
&= \frac{P(l(W, v)^\perp | l_v = j \cap v \in B_{d'}(t) \cap \text{dist}(v, u) = i) P(l_v = j) P(v \in B_{d'}(t) \cap \text{dist}(v, u) = i)}{P(l(W, v)^\perp | v \in B_{d'}(t) \cap \text{dist}(v, u) = i) P(v \in B_{d'}(t) \cap \text{dist}(v, u) = i)} \\
&= \Omega(P(l_v = j)) = \Omega(j^{-\alpha})
\end{aligned}$$

Since edges are chosen independently, the event  $l(W, B_{d'}(t))^\perp$  does not influence  $t_2$ . So  $t_2$  is a consequence from Eq. 12.

$$t_2 = P(l(u, v) | l_u = l \cap l_v = j \cap \text{dist}(u, v) = i) = \Theta \left( \frac{l \cdot j}{i \gamma} \right)$$

Replacing  $t_1$  and  $t_2$  in Eq. 11, we obtain the desired result:

$$\begin{aligned}
E(Q) &= \sum_{i=d-d'+1}^{d+d'-1} \sum_{j=a}^{\infty} t_1 \cdot t_2 \\
&= \sum_{i=d-d'+1}^{d+d'-1} \sum_{j=a}^{\infty} \Omega \left( \frac{l}{i \log n} j^{1-\alpha} \right) \\
&= \Omega \left( \sum_{i=d-d'+1}^{d+d'-1} \frac{l}{i \log n} \sum_{j=a}^{\infty} j^{1-\alpha} \right) \\
&= \Omega \left( \sum_{i=d-d'+1}^{d+d'-1} \frac{l}{i \log n} \int_a^{\infty} x^{1-\alpha} dx \right) \\
&= \Omega \left( \sum_{i=d-d'+1}^{d+d'-1} \frac{l}{i \log n} a^{2-\alpha} \right) \\
&= \Omega \left( \frac{l}{\log n} [\log(d+d'-1) - \log(d-d'+1)] a^{2-\alpha} \right)
\end{aligned}$$

This shows that the expected number of neighbors and hence the probability to have one neighbor within the desired set is indeed

$$P(l(u, V_{d'}^a) | l_u = l \cap l(W, B_{d'}(t))^\perp) = \Omega \left( \frac{l}{\log n} [\log(d+d'+1) - \log(d-d'-1)] a^{2-\alpha} \right) \quad (13)$$

as claimed.  $\square$



In the following, we model the routing process as a sequence  $X_1, X_2, \dots$ , such that  $X_i$  gives the distance of the closest neighbor of the  $i$ -th node on the path to  $t$ . The distance of the closest neighbor to  $t$  decreases in each step until a node within distance  $C$  is reached. This cannot be guaranteed for the nodes on the actual path. A node at a higher distance might be chosen if it has neighbors that are close to the destination. The monotone decrease of the sequence  $X_i$  allows us to make use of the following Lemma:

**Lemma 6.3.** *If  $X_0, X_1, \dots$  is a non-negative, integer-valued random process with  $X_0 > \lambda \geq 0$ , such that for all  $d$  with  $\lambda < d \leq X_0$*

$$P(X_{i+2} \leq d/2 | X_0, \dots, X_i = j) = \Omega\left(\frac{\log d}{\rho}\right)$$

*then the expected number of steps until the random process reduces to at most  $\lambda$  is  $\mathcal{O}(\rho \log \log X_0)$*

A proof can be found in [12], Lemma 5.2.

Let  $R_i$  denotes the set of all nodes on the path before the  $i$ -th node and their neighbors. All events need to be conditioned on the fact that no node within distance  $d = X_i$  has a link to a node in  $R_i$ , i.e. the event  $l(B_d, R_i)^\perp$ . The next result is the main part of the proof enabling the use of Lemma 6.3 with  $\lambda = e^{\log^r n}$ ,  $\rho = \log^{\alpha-r-k(3-\alpha)} n$ .

**Lemma 6.4.** *Let  $X_i$  be the distance of the closest neighbor of the  $i$ -th node on the routing path,  $1/2 \leq r \leq 1$ ,  $0 \leq k \leq \alpha - 2$ , and  $|R_i| < 1/2\sqrt{d} \log n$ . The chance that  $X_i$  is halved in the next two steps is:*

$$P\left(X_{i+2} \leq \frac{d}{2} | R_i \cap X_i = d\right) = \Omega\left(\frac{\log d \cdot \log^{r+k(3-\alpha)} n}{\log^\alpha n}\right)$$

*Proof.* Let  $u$  be the  $i$ -th node on the path. We show the result by distinguishing two cases:  $l_u < \log^k n$  and  $l_u \geq \log^k n$ . But before, a case-independent observation is made.

Note that though the distance of a neighbor of  $u$  to  $t$  is known, the distance  $\Delta$  of  $u$  is not given. We bound all the following probabilities on the event  $G = \{d + \sqrt{d} \leq \Delta \leq 2d\}$ . The first inequality is necessary to apply Lemma 6.2 with  $\frac{|R_i|}{(d+\sqrt{d}-d) \log n} < 1/2$ . The bound  $\Delta \leq 2d$  ensures that the  $\text{dist}(u, t)$  needs to be maximally quartered to have  $X_{i+2} \leq d/2$ . For a lower bound on the event  $A$  of halving the distance,  $P(A) \geq P(A|G)P(G)$  can be applied. If  $P(G) = \Omega(1)$ ,  $P(A) = \Omega(P(A|G))$  holds. It remains to show  $P(G) = \Omega(1)$ . The lower bound  $\Delta \geq d + \sqrt{d}$  holds with probability  $\Theta(1)$  by Lemma 5.2. The upper bound  $\Delta \leq 2d$  holds with probability  $\Omega(1)$  as well, as can be seen from the proof of Theorem 2.4 in [12]: The probability that an arbitrary node has a neighbor at half its distance to the destination is shown to be  $\mathcal{O}\left(\frac{1}{\log^\epsilon n}\right)$  for some  $\epsilon > 0$ . Thus, the probability of not having such a neighbor is  $\Omega\left(1 - \frac{1}{\log^\epsilon n}\right) = \Omega(1)$ , because  $\frac{1}{\log^\epsilon n} < 1/2$  for  $n$  big enough.

This concludes our case-independent observation, ensuring that indeed  $P(G) = \Omega(1)$ . In both cases,  $l_u < \log^k n$ , and  $l_u \geq \log^k n$ , we first describe an event leading to halving the distance, before formally deriving the probability of the respective event.

Assume  $l_u < \log^k n$ . The following events result in  $X_{i+2} \leq d/2$ :

- a neighbor  $v \in B_\Delta(t)$  of  $u$  has label  $l_v \geq \log^k n$ .
- $v$  has a neighbor  $w \in B_\Delta(t)$  with label  $l_w \geq \log n$
- $w$  has a link into  $B_{d/2}$
- $v$  is the node  $u$  chooses as the next node on the routing path, denote this event by  $\{Z = v\}$

All events are conditioned on  $F = l(B_d, R_i)^\perp \cap l_u \leq \log^k n \cap G$ . Formally, the probability is

determined by:

$$\begin{aligned}
& P\left(l(u, V_{\Delta}^{\log^k n}) \cap l(v, V_{\Delta}^{\log n}) \cap l(w, B_{d/2}) \cap Z = v | F\right) \\
&= P\left(l(u, V_{\Delta}^{\log n}) | F\right) \cdot P\left(l(v, V_{\Delta}^{\log n}) | l(u, V_{\Delta}^{\log^k n}) \cap F\right) \\
&\quad \cdot P\left(l(w, B_{d/2}) | l(v, V_{\Delta}^{\log n}) \cap l(u, V_{\Delta}^{\log^k n}) \cap F\right) \\
&\quad \cdot P\left(Z = v | l(w, B_{d/2}) \cap l(v, V_{\Delta}^{\log n}) \cap l(u, V_{\Delta}^{\log^k n}) \cap F\right) \\
&:= q_1 q_2 q_3 q_4
\end{aligned} \tag{14}$$

We now subsequently bound  $q_1$ ,  $q_2$ ,  $q_3$  and  $q_4$ .  $q_1$  can be derived using Lemma 6.2 with  $d = d' = \Delta$  and the fact that the probability of having a link is minimal for a node  $u$  with  $l_u = 1$ .

$$\begin{aligned}
q_1 &= P\left(l(u, V_{\Delta}^{\log^k n}) | l(B_d, R_i)^{\perp} \cap l_u \leq \log^k n \cap G\right) \\
&\geq P\left(l(u, V_{\Delta}^{\log^k n}) | l(B_d, R_i)^{\perp} \cap l_u = 1 \cap G\right) \\
&= \Omega\left(\frac{1}{\log n} [\log(2\Delta - 1) - 0] \log^{k(2-\alpha)} n\right) \\
&= \Omega\left(\frac{\log d}{\log n} \log^{k(2-\alpha)} n\right)
\end{aligned}$$

The last step uses  $d \leq \Delta \leq 2d$ . Since links are selected independently, the events  $l(u, V_{\Delta}^{\log^k n})$  and  $l(v, V_{\Delta}^{\log n})$  are independent, but  $v \in V_{\Delta}^{\log^k n}$  influences  $l(v, V_{\Delta}^{\log n})$ . The maximal distance  $v$  can have to  $t$  is  $\Delta$ . Because labels are selected independently  $l_u \leq \log^k n$  does not influence  $q_2$  or  $q_3$ . Hence  $q_2$  is derived similarly to  $q_1$ :

$$\begin{aligned}
q_2 &= P\left(l(v, V_{\Delta}^{\log n}) | l(B_d, R_i)^{\perp} \cap l_v \geq \log^k n \cap G\right) \\
&\geq P\left(l(v, V_{\Delta}^{\log n}) | l(B_d, R_i)^{\perp} \cap l_v = \log^k n \cap G\right) \\
&= \Omega\left(\frac{\log^k n}{\log n} [\log(2\Delta - 1) - 0] \log^{(2-\alpha)} n\right) \\
&= \Omega\left(\frac{\log d}{\log n} \log^k n \log^{(2-\alpha)} n\right)
\end{aligned}$$

Note that  $B_{d/2} = V_{d/2}^1$ , so Lemma 6.2 is applied to determine  $q_3$  as well. Furthermore, the function  $\log(x + d/2) - \log(x - d/2)$ , being a monotone decreasing function for  $x > d/2$ , assumes its maximum in the interval  $[d, 2d]$  at  $\Delta = d$ .

$$\begin{aligned}
q_3 &= P\left(l(w, B_{d/2}) | l(B_d, R_i)^{\perp} \cap l_w \geq \log n \cap G\right) \\
&\geq P\left(l(w, B_{d/2}) | l(B_d, R_i)^{\perp} \cap l_w = \log n \cap G\right) \\
&= \Omega\left(\frac{\log n}{\log n} [\log(\Delta + d/2 - 1) - \log(\Delta - d/2 + 1)]\right) \\
&= \Omega(\log(d + d/2 - 1) - \log(d - d/2 + 1)) \\
&= \Omega(1)
\end{aligned}$$

We are considering the case when  $u$  has  $\mathcal{O}(\log^k n)$  neighbors at distance at least  $d$  from  $t$ . Note that the probability that one of  $\log^k n$  arbitrary nodes link to a certain node is asymptotically the same as considering one node with degree  $\log^k n$ . Hence the probability that  $v$  or any one of  $u$ 's remaining neighbors has the closest neighbor to  $t$  are of the same order, i.e.  $q_4 = \Omega(1)$ .

Combining the results for the individual terms, we get a bound for halving the distance in case of  $l_u \leq \log^k n$ .

$$\begin{aligned}
P(X_{i+2} \leq \frac{d}{2} | R_i \cap X_i = d) \\
&= \Omega(q_1 q_2 q_3 q_4) \\
&= \Omega\left(\frac{\log d}{\log n} \log^{k(2-\alpha)} n \cdot \frac{\log d}{\log n} \log^k n \log^{(2-\alpha)} n\right) \\
&= \Omega\left(\frac{\log d \log^{k(3-\alpha)} n}{\log^\alpha n} \log d\right) \\
&= \Omega\left(\frac{\log d \log^{k(3-\alpha)} n}{\log^\alpha n} \log^r n\right)
\end{aligned}$$

The last step uses that  $\log d > \log e^{\log^r n} = \log^r n$ . Having shown the result for the case  $l_u < \log^k n$ ,  $l_u \geq \log^k n$  has to be treated differently, because  $q_4$  cannot be bounded as above. However, this case is easier, since one does not need to contact a node with label at least  $\log^k n$  first. Here we consider the following events:

- $u$  has a neighbor  $w$  within distance  $\Delta$  of  $t$  with degree at least  $\log n$
- $w$  links into  $B_{d/2}$

Since  $u$  knows the identifiers of  $w$ 's neighbors, it will select  $w$ , unless there is some other node being both a neighbor to  $u$  and a node in  $B_{d/2}$ . This corresponds to the second and third events in case  $l_u \leq \log^k n$  and are already bounded by  $q_2$  and  $q_3$ . Formally, this event can be written as follows:

$$\begin{aligned}
&P\left(l(u, V_\Delta^{\log n}) \cap l(w, B_{d/2}) | l_u \geq \log^k n \cap l(B_d, R_i)^\perp \cap G\right) \\
&= P\left(l(u, V_\Delta^{\log n}) | l_u \geq \log^k n \cap l(B_d, R_i)^\perp \cap G\right) \\
&\quad \cdot P\left(l(w, B_{d/2}) | l(u, V_\Delta^{\log n}) \cap l_u \geq \log^k n \cap l(B_d, R_i)^\perp \cap G\right) \\
&= q_2 q_3 = \Omega\left(\frac{\log d}{\log n} \log^k n \log^{(2-\alpha)} n\right)
\end{aligned}$$

So, we can half the distance in one step with probability  $\Omega\left(\frac{\log d}{\log n} \log^k n \log^{(2-\alpha)} n\right)$ . The sequence  $X_i$  is decreasing and  $q_1 q_4 \leq 1$ , so in case  $l_u \geq \log^k n$ , it holds that

$$P(X_{i+2} \leq \frac{d}{2} | R_i \cap X_i = d) = \Omega\left(\frac{\log d \log^{k(3-\alpha)} n}{\log^\alpha n} \log^r n\right)$$

as well. This completes the proof.  $\square$

Based on Lemma 6.3 and Lemma 6.4,  $\mathbb{E}(R_1^{\text{NoN}}(s, t))$  can be bounded.

**Lemma 6.5.** *For a graph  $G = (V, E) \in \mathcal{D}(n, 1, C, S_\alpha)$  with  $C < e^{\log^{1/2} n}$ , two nodes  $s, t \in V$ ,  $1/2 \leq r \leq 1$  and  $0 \leq k \leq \alpha - 2$ , the expected routing length of NextBestOnce-NoN to reach a node within distance  $e^{\log^r n}$  of  $t$  is*

$$\mathbb{E}(R_1^{\text{NoN}}(s, t)) = \mathcal{O}\left(\log^{\alpha-r-k(3-\alpha)} n \log \log n\right). \quad (15)$$

*Proof.* By Lemma 6.4 the probability to half the distance during the next two steps is given by

$$P(X_{i+2} \leq \frac{d}{2} | X_1, X_2, \dots, X_i = d) = \Omega\left(\frac{\log d \cdot \log^{r+k(3-\alpha)} n}{\log^\alpha n}\right) = \Omega\left(\frac{\log d}{\log^{\alpha-r-k(3-\alpha)} n}\right)$$

as long as  $d > e^{\log^r n}$  and  $|R_i| < \sqrt{d}/2$ . The later holds with probability at least  $1 - \frac{1}{n}$ , as can be seen from the proof for the upper bound of *NextBestOnce* (see [26] or for a similar

argumentation [12], Theorem 2.4). It is shown that *NextBestOnce* needs at most  $\mathcal{O}(\log^3 n)$  steps with probability  $\Omega(1 - 1/n)$ . Since we assume the maximal degree to be bounded logarithmically,  $|R_i| \leq K \log^4 n$  for some constant  $K$  follows. Hence, Lemma 6.3 with  $\rho = \log^{\alpha-r-k(3-\alpha)} n$  and  $\lambda = e^{\log^r b}$  can be applied to obtain Eq. 15:

$$\begin{aligned}
& \mathbb{E}(R_1^{NoN}(s, t)) \\
&= P(|R_i| \leq K \log^4 n) \mathbb{E}(R_1^{NoN}(s, t) | |R_i| \leq K \log^4 n) \\
&+ (1 - P(|R_i| \leq K \log^4 n)) \mathbb{E}(R_1^{NoN}(s, t) | |R_i| > K \log^4 n) \\
&= \Omega\left(1 - \frac{1}{n}\right) \mathcal{O}\left(\log^{\alpha-r-k(3-\alpha)} n \log \log n\right) + \mathcal{O}\left(\frac{1}{n}\right) \mathcal{O}(n) \\
&= \mathcal{O}\left(\log^{\alpha-r-k(3-\alpha)} n \log \log n\right)
\end{aligned}$$

In the second last step holds since  $e^{\log^r n} > C$ , so the distance is guaranteed to decrease in each step, and thus maximally  $n/2 - e^{\log^r n}$  are needed to complete the first phase.  $\square$

Theorem 3.3 can now be shown solving a two-dimensional extremal value problem.

*Proof.* It follows from Lemma 6.5 and 6.1 that for all  $(k, r) \in [0, \alpha - 2] \times [1/2, 1]$

$$\mathbb{E}(R^{NoN}(s, t)) = \mathcal{O}\left(\log^{\alpha-r-k(3-\alpha)} n \log \log n + \log^{r(\alpha-1)} n \log \log n + C^3 \log n\right)$$

We need to find  $(k_{min}, r_{min})$  such that

$$f(k, r) = \log^{\alpha-r-k(3-\alpha)} n + \log^{r(\alpha-1)} n \quad (16)$$

is minimized. Computing the gradient of  $f$  gives:

$$Df = \begin{pmatrix} -(3-\alpha) \log \log n (\log n)^{\alpha-r_{min}-k_{min}(3-\alpha)} \\ -\log \log n (\log n)^{\alpha-r_{min}-k_{min}(3-\alpha)} + (\alpha-1) \log \log n (\log n)^{r_{min}(\alpha-1)} \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

So the  $f$  takes its minimum on the border of the  $[0, \alpha - 2] \times [1/2, 1]$ , e.g. if either  $k_{min} = 0$ ,  $k_{min} = \alpha - 2$ ,  $r_{min} = 0.5$  or  $r_{min} = 1$ . When  $k_{min} = 0$ , a node of degree at least one needs to be contacted first. This leads essentially to the same scenario used to obtain the bound for *NextBestOnce*, and cannot have an improved complexity. The same goes for the case  $r_{min} = 1$ , because  $e^{\log^1 n} = n$ , so only the second phase, for which the complexity is bounded by that of *NextBestOnce* is considered. As for  $r_{min} = 0.5$ , observe the exponent of the first summand of  $f$ .

$$\alpha - 0.5 - k \cdot (3 - \alpha) \geq \alpha - 0.5 - (\alpha - 2) \cdot (3 - \alpha) > \alpha - 1$$

The last step uses that  $2 < \alpha < 3$ . So, an improved bound with regard to *NextBestOnce* can only be obtained for  $k_{min} = \alpha - 2$ . We determine  $r_{min}$  by minimizing

$$g(r) = (\log n)^{\alpha-(\alpha-2)(3-\alpha)-r} + (\log n)^{r(\alpha-1)}$$

The first derivative of  $g$  is

$$g'(r) = -\log \log n (\log n)^{\alpha-(\alpha-2)(3-\alpha)-r} + (\alpha-1) \log \log n (\log n)^{r(\alpha-1)}$$

Setting  $g'(r_{min}) = 0$  we get that

$$\begin{aligned}
(\alpha-1) \log \log n (\log n)^{r_{min}(\alpha-1)} &= \log \log n (\log n)^{\alpha-(\alpha-2)(3-\alpha)-r_{min}} \\
(\log n)^{r_{min}(\alpha-1)+r_{min}} &= \frac{1}{\alpha-1} (\log n)^{\alpha-(\alpha-2)(3-\alpha)} \\
((\log n)^{r_{min}})^\alpha &= \frac{1}{(\alpha-1)} (\log n)^{\alpha-(\alpha-2)(3-\alpha)} \\
(\log n)^{r_{min}} &= \frac{1}{(\alpha-1)^{1/\alpha}} (\log n)^{1-\frac{(\alpha-2)(3-\alpha)}{\alpha}}
\end{aligned}$$

Finally, we get

$$\begin{aligned} r_{min} &= \frac{\log \frac{1}{(\alpha-1)^{1/\alpha}} + \left(1 - \frac{(\alpha-2)(3-\alpha)}{\alpha}\right) \log \log n}{\log \log n} \\ &= \frac{\log \frac{1}{(\alpha-1)^{1/\alpha}}}{\log \log n} + \left(1 - \frac{(\alpha-2)(3-\alpha)}{\alpha}\right) \end{aligned} \quad (17)$$

This is indeed a minimum since

$$g''(r_{min}) = (\log \log n)^2 \left( (\log n)^{\alpha - (\alpha-2)(3-\alpha) - r_{min}} + (\alpha-1)^2 (\log n)^{r_{min}(\alpha-1)} \right) > 0$$

Consider that for  $a = \log \frac{1}{(\alpha-1)^{1/\alpha}}$

$$(\log n)^{\frac{a}{\log n}} = 2^{\frac{a \log n}{\log n}} = 2^a.$$

By this, the first summand in Eq. 17 does not contribute to the asymptotic complexity, allowing us to use only the second summand

$$r_{min}^* = 1 - \frac{(\alpha-2)(3-\alpha)}{\alpha}$$

for the routing bound in Theorem 3.3. The upper bound on *NextBestOnce-NoN* is then obtained as

$$\mathcal{O}(f(k_{min}, r_{min}^*) \log \log n + C^3 \log n) = \mathcal{O}\left(\log^{\delta(\alpha)(\alpha-1)} n \log \log n + C^3 \log n\right)$$

for  $\delta(\alpha) = 1 - \frac{(\alpha-2)(3-\alpha)}{\alpha}$ . This completes the remaining steps in the proof of Theorem 3.3.  $\square$

## 7 Conclusion and Future Work

We have provided an analysis of *NextBestOnce*, an alternative routing algorithm based solely on information about direct neighbors with guaranteed convergence in any embedded graph. In the context of a model for heuristically embedded social graphs, the expected routing length of *NextBestOnce* can be bound in terms of the number  $n$  of participants, the exponent  $\alpha$  of the scale-free degree distribution of the social graph, and a parameter  $C$  measuring the accuracy of the embedding: *NextBestOnce*'s expected routing length satisfies the upper and lower bounds  $\mathcal{O}(\log^{\alpha-1} n \log \log n + C^3 \log n)$  and  $\Omega(\log^{\alpha-1} n + C)$ . By this, *NextBestOnce* achieves polylog performance as long as  $C$  is bound polylog. This result complements our earlier work, which shows that currently deployed algorithms do not achieve polylog performance [28], even in case of constant  $C$ . Furthermore, we have shown that using information about the two-hop neighborhood indeed achieves an asymptotically decreased routing length for sufficiently accurate embeddings: The extended algorithm *NextBestOnce-NoN* needs  $\mathcal{O}(\log^{\delta(\alpha)(\alpha-1)} n \log \log n + C^3 \log n)$  hops for  $\delta(\alpha) < 1$ .

The price of the increased performance are additional local computation, storage, and maintenance costs due to the increased number of identifiers considered for routing decisions. The expected number of two-hop neighbors is given by  $E(S_\alpha^2) = \Theta(\mu^{3-\alpha})$ , where  $\mu$  is the maximal degree, whereas the expected number of neighbors  $E(S_\alpha)$  is bound by a constant. In other words, the costs per hop are constant when using only direct neighborhood information, but increase with the maximal degree when considering the two-hop neighborhood. Note that a logarithmic maximal degree is sufficient for the proof of Theorem 3.3. Assuming a logarithmic maximal degree, the expected costs are polylog in the number of participants, as is common in other structured overlays such as DHTs. With that in mind, the additional costs seem a reasonable price for the significantly shorter routes *NextBestOnce-NoN* offers, especially when considering that routing takes often hundreds of hops in current Darknet implementations.

Since our aim was to show the superiority of *NextBestOnce-NoN* in comparison to *NextBestOnce*, we did not provide a lower bound on the performance of *NextBestOnce-NoN*. NoN routing has been shown to be optimal in a similar context [20], in as far as that the expected routing length is asymptotically equal to the diameter of the graph. It remains to be seen if the result holds in case of scale-free degree distributions as well. In addition, we plan to analyze the dependence of routing length and the accuracy of the embedding in more detail, aiming to close the gap between the linear lower and the at least cubic upper bound.

## References

- [1] Sonja Buchegger, Doris Schiöberg, Le Hung Vu, and Anwitaman Datta. PeerSoN: P2P Social Networking. In *Social Network Systems*, 2009.
- [2] Augustin Chaintreau, Pierre Fraigniaud, and Emmanuelle Lebhar. Networks become navigable as nodes move and forget. In *Proceedings of the 35th international colloquium on Automata, Languages and Programming, ICALP '08*, 2008.
- [3] Ian Clarke, Oskar Sandberg, Matthew Toseland, and Vilhelm Verendel. Private communication through a network of trusted connections: The dark freenet. <http://freenetproject.org/papers.html>, 2010.
- [4] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *International Workshop on Design Issues in Anonymity and Unobservability*, 2000.
- [5] Don Coppersmith, David Gamarnik, and Maxim Sviridenko. The diameter of a long-range percolation graph. *Random Struct. Algorithms*, 21(1), 2002.
- [6] Leucio-Antonio Cutillo, Refik Molva, and Thorsten Strufe. Privacy Preserving Social Networking Through Decentralization. In *6th International Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 145 – 152, 2009.
- [7] Andrej Cvetkovski and Mark Crovella. Hyperbolic embedding and routing for dynamic graphs. In *Proceedings of the 28th IEEE International Conference on Computer Communications, INFOCOM '09*, 2009.
- [8] Matteo Dell’Amico. Mapping small worlds. In *Proceedings of the 7th International Conference on Peer-to-Peer Computing, P2P '07*, 2007.
- [9] David Eppstein and Michael T. Goodrich. Succinct greedy geometric routing using hyperbolic geometry. *IEEE Trans. Computers*, 60(11):1571–1580, 2011.
- [10] Nathan S. Evans and Christian Grothoff. R5N: Randomized recursive routing for restricted-route networks. In *Proceedings of the 5th International Conference on Network and System Security, NSS '11*, 2011.
- [11] Roland Flury, Sriram V. Pemmaraju, and Roger Wattenhofer. Greedy routing with bounded stretch. In *Proceedings of the 28th IEEE International Conference on Computer Communications, INFOCOM '09*, 2009.
- [12] Pierre Fraigniaud and George Giakkoupis. The effect of power-laws on the navigability of small worlds. In *Proceedings of the 23rd annual ACM symposium on Principles of distributed computing, PODC '09*, 2009.
- [13] Pierre Fraigniaud and George Giakkoupis. On the searchability of small-world networks with arbitrary underlying structure. In *Proceedings of the 42nd ACM symposium on Theory of computing, STOC '10*, 2010.
- [14] George Giakkoupis and Nicolas Schabanel. Optimal path search in small worlds: dimension matters. In *Proceedings of the 43rd Symposium on Theory of Computing, STOC '11*, 2011.
- [15] Julien Herzen, Cédric Westphal, and Patrick Thiran. Scalable routing easy as pie: A practical isometric embedding protocol. In *Proceedings of the 19th IEEE International Conference on Network Protocols, ICNP '11*, 2011.

- [16] Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas E. Anderson. Privacy-preserving p2p data sharing with oneswarm. In *Proceedings of the ACM SIGCOMM 2010 conference, SIGCOMM '10*, 2010.
- [17] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the 32nd Symposium on Theory of Computing, STOC '00*, 2000.
- [18] Robert Kleinberg. Geographic routing using hyperbolic space. In *Proceedings of the 26th IEEE International Conference on Computer Communications, INFOCOM '07*, 2007.
- [19] Emmanuelle Lebhar and Nicolas Schabanel. Almost optimal decentralized routing in long-range contact networks. In *Proceedings of the 30th international colloquium on Automata, Languages and Programming, ICALP '04*, 2004.
- [20] Gurmeet Singh Manku, Moni Naor, and Udi Wieder. Know thy neighbor's neighbor: the power of lookahead in randomized p2p networks. In *Proceedings of the 36th annual ACM symposium on Theory of computing, STOC '04*, 2004.
- [21] C. Martel and V. Nguyen. The complexity of message delivery in kleinberg's small-world model. Technical report, UC Davis Department of Computer Science, 2003.
- [22] Chip Martel and Van Nguyen. Analyzing kleinberg's (and other) small-world models. In *Proceedings of the 33rd annual ACM symposium on Principles of distributed computing, PODC '04*, 2004.
- [23] Petar Maymounkov. Greedy embeddings, trees, and euclidean vs. lobachevsky geometry. <https://www.pdos.lcs.mit.edu/~petar/papers/maymounkov-greedy-prelim.pdf>, 2006.
- [24] Prateek Mittal, Matthew Caesar, and Nikita Borisov. X-vine: Secure and pseudonymous routing using social networks. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium, NDSS '12*, 2012.
- [25] Bogdan C. Popescu, Bruno Crispo, and Andrew S. Tanenbaum. Safe and private data sharing with turtle: Friends team-up and beat the system. In *Proceedings of the 12th International Workshop Security Protocols*. Springer, 2006.
- [26] Stefanie Roos. Analysis of routing in sparse small-world topologies. Diplomarbeit, TU Darmstadt, 2011.
- [27] Stefanie Roos and Thorsten Strufe. Provable polylog routing for darknets. In *Proceedings of the 4th Workshop on Hot Topics in Peer-to-peer Computing and Online Social Networking, HotPOST '12*, 2012.
- [28] Stefanie Roos and Thorsten Strufe. A contribution to darknet routing. In *Proceedings of the 32nd IEEE International Conference on Computer Communications, INFOCOM '13*, 2013.
- [29] Oskar Sandberg. Distributed routing in small-world networks. In *Proceedings of the 8th Workshop on Algorithm Engineering and Experiments, ALENEX '06*, 2006.
- [30] Benjamin Schiller, Stefanie Roos, Andreas Höfer, and Thorsten Strufe. Attack resistant network embeddings for darknets. In *Proceedings of the 30th Symposium on Reliable Distributed Systems Workshops, SRDSW '11*, 2011.
- [31] Eugene Vasserman, Rob Jansen, James Tyra, Nicholas Hopper, and Yongdae Kim. Membership-concealing overlay networks. In *Proceedings of the 17th ACM conference on Computer and communications security, CCS '09*, 2009.
- [32] Cédric Westphal and Guanhong Pei. Scalable routing via greedy embedding. In *Proceedings of the 28th IEEE International Conference on Computer Communications, INFOCOM '09*, 2009.